

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра автоматизації та управління в технічних системах**

«До захисту допущено»

Завідувач кафедри

_____ О.І. Ролік

«__» _____ 2019 р.

**Дипломний проект
на здобуття ступеня бакалавра
з напрямку підготовки 6. 050201 «Системна інженерія»
на тему: «Телеграм-бот для взаємодії викладачів та студентів»**

Виконав:

студент IV курсу, групи ІА-52

Любін Ігор Миколайович _____

Керівник:

Старший викладач, Март Б. А. _____

Рецензент:

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць
інших авторів без відповідних
посилань.

Студент (-ка) _____

Київ – 2019 рік

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки – 6. 050201 «Системна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.І. Ролік

« ____ » _____ 2019 р.

ЗАВДАННЯ
на дипломний проект студенту
Любін Ігорю Миколайовичу

1. Тема проекту «Telegram-бот для взаємодії викладачів та студентів», керівник проекту Март Богдан Анатолійович, старший викладач, затверджені наказом по університету від « ____ » _____ 2019 р. № _____
2. Термін подання студентом проекту 19.06.2019 р.
3. Вихідні дані до проекту Telegram, операційна система Linux, система взаємодії
4. Зміст пояснювальної записки Вступ, опис об'єкта проектування, огляд існуючих рішень, реалізація проекту, архітектура серверної частини, інструкція користувача, висновки
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) Діаграма класів, Блок-схема роботи Telegram-бота, UML-діаграма, Структура бази даних
7. Дата видачі завдання 20 березня 2019 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Аналіз предметної області	29.03.2019	
2	Аналіз об'єкта проектування	05.04.2019	
3	Огляд існуючих рішень	19.04.2019	
4	Реалізація проекту	24.05.2019	
5	Оформлення документації проекту	14.06.2019	

Студент

І. М. Любін

Керівник проекту

Б. А. Март

АНОТАЦІЯ

Любін І. М. Телеграм-бот для взаємодії викладачів та студентів. КПП ім. Ігоря Сікорського, Київ, 2019.

Проект містить 64 сторінки тексту, 15 літературних джерел, 4 конструкторських документа.

Ключові слова: back end, веб-сервіс, месенджер, навчальний процес, педагогічне навантаження, система взаємодії, телеграм-бот.

Об'єктом дослідження є навчальний процес у вищих навчальних закладах.

Метою дипломного проекту є розробка методів для спрощення взаємодії викладачів та студентів під час навчального процесу.

У дипломному проекті розроблено систему керування Телеграм-ботом. Він призначений для спілкування викладачів та студентів, поліпшення процесу навчання та поширеності навчальних матеріалів. Проведено аналіз подібних систем, виконаних з використанням подібних технологій та архітектури. Описана причина вибору використаних технологій та їх переваги. Значну увагу приділено використанню Telegram Bot API, описано методи, їх параметри та спосіб використання.

SUMMARY

Lubin I. M. Telegram Bot for «teacher-student» interaction. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2019.

The project contains 63 pages. text, links to 15 literary sources, 4 design documents.

Keywords: back end, web service, messenger, learning process, pedagogical load, interaction system, telegram bot.

The object of research is the educational process in higher educational institutions.

The purpose of the diploma project is to develop methods to facilitate the interaction of teachers and students during the educational process.

The diploma project has developed a telegram bot control system. It is intended for communication of teachers and students, improving the learning process and the prevalence of teaching materials. An analysis of similar systems performed using similar technologies and architecture has been carried out. The reason for choosing the technologies used and their benefits is described. Considerable attention is paid to using the Telegram Bot API, describes the methods, their parameters and how to use them.

**Пояснювальна записка
до дипломного проекту
на тему: «Телеграм-бот для взаємодії
викладачів та студентів»**

Київ – 2019 рік

ЗМІСТ

ЗМІСТ	2
ВСТУП	5
1 ОПИС ОБ'ЄКТА ПРОЕКТУВАННЯ	7
1.1 Вибір технічних засобів	8
1.1.1 Реалізація у вигляді веб-сервісу	8
1.1.2 Реалізація у вигляді веб-сервісу	8
1.1.3 Розгляд месенджерів	8
1.1.4 Розгляд варіантів вибору технологій	10
1.2 Структура системи	11
1.3 Функціональні частини системи	12
1.3.1 Реєстрація	13
1.3.2 Управління матеріалами	13
1.3.3 Отримання інформації про користувача	13
1.3.4 Отримання інформації про групи користувачів	14
1.3.5 Додавання користувацьких чатів	14
1.3.6 Адміністрування системи безпосередньо в самій системі	14
Висновки до розділу 1	15
2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	16
2.1 Подібні системи взаємодії	16
2.2 Онлайн курси «Skylang» для вивчення іноземних мов	18
2.3 Веб-сервіс з онлайн курсами різної направленості «L-a-b-a»	20
2.4 Електронний кампус	21
2.5 Телеграм-бот «Предложка лайва»	21
2.6 Телеграм-бот «ЦФВС КПІ ім. Ігоря Сікорського»	22
2.7 Телеграм-бот «KPI schedule»	22
Висновки до розділу 2	24
3 РЕАЛІЗАЦІЯ ПРОЕКТУ	25
3.1 Опис розробленого функціоналу	25
3.1.1 Реєстрація	25

3.1.2 Інтерактивне меню	27
3.1.3 Додавання користувацьких чатів	27
3.1.4 Автоматизовані опитування	28
3.2 Опис запитів до Telegram Bot API	28
3.2.1 Типи зворотного зв'язку	28
3.2.2 Метод «getUpdates»	29
3.2.3 Метод «setWebhook»	30
3.2.4 Метод «getWebhookInfo»	31
3.2.5 Метод «sendMessage»	31
3.2.6 Метод «sendContact»	32
3.2.7 Метод «sendChatAction»	33
3.2.8 Метод «sendPhoto»	33
3.2.9 Метод «getUserProfilePhotos»	34
3.2.10 Метод «getFile»	34
3.2.11 Метод «exportChatInviteLink»	35
3.2.12 Метод «kickChatMember»	35
3.2.13 Метод «getChat»	36
3.2.14 Метод «deleteMessage»	36
3.2.15 Метод «editMessageCaption»	37
3.2.16 Метод «editMessageText»	37
Висновки до розділу 3	39
4 АРХІТЕКТУРА СЕРВЕРНОЇ ЧАСТИНИ	40
4.1 Технології використані при розробці проекту	40
4.1.1 Веб-сервер Apache2	40
4.1.2 Мова програмування PHP 7.2	41
4.1.3 Фреймворк PHP Laravel 5.7	43
4.1.4 Реляційна база даних MySQL	44
4.1.5 Веб-додаток phpMyAdmin	45
4.1.6 Нереляційна база даних Redis	46
4.1.7 Ubuntu 16.04	47
4.1.8 Програмний пакет ngrok	48
4.1.9 Пакетний менеджер Composer	48

4.1.10 Система контролю версій Git	49
4.2 Серверна архітектура	49
Висновки до розділу 4	51
5 ІНСТРУКЦІЯ КОРИСТУВАЧА	52
5.1 Процес реєстрації	52
5.2 Додавання матеріалів	54
5.3 Отримання інформації про користувача	54
5.4 Додавання користувацьких чатів	55
5.5 Отримання матеріалів	55
Висновки до розділу 5	56
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	59
ДОДАТОК А	61
ДОДАТОК Б	63
ДОДАТОК В	66
ДОДАТОК Г	67

ВСТУП

На сьогоднішній день, вища освіта набула великої популярності в порівнянні з минулим століттям. Стрімко зросли такі показники як: кількість вищих навчальних закладів, їх масштабність (з точки зору кількості інститутів та факультетів), різноманітності спрямувань освітніх програм, а також кількість студентів, які навчаються у вищих навчальних закладах. Через такі глобальні зміни в масштабності виникла потреба в залученні більшої кількості викладачів та персоналу університетів в цілому.

Через стрімкий ріст рівня телекомунікаційних технологій, процес спілкування переноситься в середовище онлайн. Зростає популярність систем для взаємодії, месенджерів, соціальних мереж, веб-сервісів та інших. З огляду на зручність такого способу взаємодії, є сенс впровадження веб-технологій у навчальний процес. Це допоможе зменшити навантаження та поліпшити ефективність самого процесу.

Актуальність дипломного проекту полягає в тому, що взаємодія викладачів та студентів ускладнена їх великою кількістю. Викладачі не в змозі повноцінно взаємодіяти зі студентами через що, в свою чергу, створюються складнощі для студентів, а саме через складність зв'язку з викладачем. Ці фактори спонукають до розробки зручної та простої системи спілкування та взаємодії викладачів пов'язаної з навчальним процесом.

Об'єктом дослідження є навчальний процес у вищих навчальних закладах.

Предметом дослідження є методи та способи спрощення та покращення навчального процесу в вищих навчальних закладах.

Метою дипломного проекту є розробка методів для спрощення взаємодії викладачів та студентів під час навчального процесу.

Головне завдання - спрощення взаємодії викладачів та студентів під час навчального процесу. Для досягнення мети було поставлено такі задачі:

					IA52.160БАК.002 ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підп.	Дата		5

- провести аналіз схожих систем, які могли б вирішувати поставлене завдання;
- провести аналіз технологій для розробки дипломного проекту;
- обрати поєднання технологій для швидкодії та стійкості системи;
- розробити частину функціонального елементу серверної архітектури сервісу.

Практичне значення одержаних результатів:

- розробити Телеграм-бота для спрощення взаємодії викладачів та студентів під час навчального процесу.

Бакалаврський проект містить: вступ, 5 основних розділів, висновки, список використаних джерел із 15 найменувань, 4 додатків. графічна частина включає 4 креслеників формату А3 . Загальний обсяг роботи 64 сторінки.

					ІА52.160БАК.002 ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підп.	Дата		6

1 ОПИС ОБ'ЄКТА ПРОЕКТУВАННЯ

Розроблювана система ставить перед собою за мету спрощення навчального процесу за рахунок автоматизації деяких монотонних та часто повторюваних процесів, які виконуються як студентами так і викладачами. До цих процесів відносяться такі як: збір інформації про студентів та викладачів, видача та отримання учбових матеріалів стосовно учбового процесу, швидкого отримання інформації про студента чи викладача та інші.

Економічна складова також грає не меншу роль, оскільки метою проектування даної системи є не лише поліпшення та спрощення процесів взаємодії викладачів та студентів в навчально-виховних заходах вищих навчальних закладів, а й простота підтримки даної системи в майбутньому та дешевизна її використання. Економічний аспект системи є важливим, оскільки добре виконана, але дорога система буде користуватись малою популярністю, бо на ринку завжди будуть існувати дешевші або безкоштовні варіанти реалізації подібного функціоналу.

Останньою важливою характеристикою є стійкість та надійність розроблюваної системи. Ці характеристики обумовлені тим фактором, що розроблювана система створюється задля пришвидшення та спрощення навчально-виховних заходів та процесів.

В свою чергу нестійка або нестабільна система, в будь-якому сенсі, створюватиме складнощі в її використанні. Така система ставила б під сумнів працездатність розроблюваної системи, тому ці характеристики є важливими.

1.1 Вибір технічних засобів

З огляду на поставлене завдання та можливі варіанти його реалізації, були розглянуті та обрані декілька варіантів.

1.1.1 Реалізація у вигляді веб-сервісу

Даний варіант реалізації являє собою класичний веб-сервіс. В своєму складі він має користувацький інтерфейс (Front End частина сервісу), який виконується за допомогою технологій JavaScript, HTML та CSS. Також він має серверну частину (Back End частина сервісу), яка містить всю бізнес логіку сервісу, засоби взаємодії з базою даних та власне сам веб-проект. Після детальної оцінки данного варіанту реалізації, даний варіант був відкинутий. Причиною цьому послуговували декілька факторів, таких як не виправдана об'ємність процесу розробки проекту, знижена зручність використання даного проекту за призначенням, через необхідність використання його в браузері та більш низька загальна швидкість роботи в порівнянні з обраним варіантом технічної реалізації проекту.

1.1.2 Реалізація у вигляді веб-сервісу

Даний варіант розробки відзначився більш підходящою основою для розроблюваної системи, оскільки він має кілька переваг. По-перше, використання месенджера прибирає потребу розробки якогось інтерфейсу, оскільки він вже реалізований в самому месенджері. По-друге, це підвищує швидкодію сервісу.

1.1.3 Розгляд месенджерів

					IA52.160БАК.002 ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підп.	Дата		8

Для даного проекту розглядалася можливість використання таких месенджерів:

- WhatsApp;
- Viber;
- Телеграм (ориг. Telegram).

Серед розглянутих месенджерів кожен із варіантів має свої переваги та недоліки. Слід додати що принцип роботи з кожним із месенджерів схожий. Відмінність у їх використанні обумовлена лише економічною та технічно-функціональною частиною кожного сервісу.

Першим розглядався месенджер WhatsApp та його API WhatsApp Business. До уваги були взяті такі фактори:

- даний месенджер наразі не є досить популярним;
- API, створене компанією Facebook, є платним;
- порівняно менша кількість функціоналу самого месенджера.

Через ці фактори даний месенджер не був обраний для розробки сервісу.

Наступним на черзі до розгляду став месенджер Viber та його Viber API. Слід відмітити, що даний месенджер був і залишається досить популярним серед користувачів. На сьогоднішній день кількість його користувачів складає більше одного мільярда. Але він має і недоліки, серед яких:

- мала функціональність самого месенджера;
- повільна робота месенджера в порівнянні з конкурентами;
- складність переходу з одного пристрою на інший, складно зберегти дані переписок, що в свою чергу ускладнить використання розроблюваного сервісу;
- платне використання функціоналу.

Вище перераховані фактори не задовольняють поставлені вимоги до системи, тому цей месенджер також не був використаний в розробці сервісу.

Останнім до розгляду став месенджер Телеграм та його Telegram Bot API. З часів появи даного месенджера і до сьогоднішнього дня, він здобув немалу популярність у всьому світі. Користувачі найчастіше обирають його через швидкість роботи, зручність та простоту інтерфейсу, і не дивлячись на це, доволі велика варіативність функціоналу. З його слабких сторін можна відмітити лише таку - відсутність функціоналу видалення повідомлень через Telegram Bot API. З огляду на це можна зробити висновок, що даний недолік не є критичним і в цілому даний месенджер задовольняє вимоги даного проекту.

1.1.4 Розгляд варіантів вибору технологій

Також потрібно обрати відповідні технології, на базі яких буде вестися розробка. Розглядалися такі технології для розробки проекту:

- веб-сервер Apache2;
- веб-сервер Nginx;
- мова програмування PHP 7.2;
- фреймворк PHP Laravel 5.7;
- фреймворк PHP Yii 2;
- реляційна база даних MySQL;
- реляційна база даних PostgreSQL;
- реляційна база даних MS SQL;
- реляційна база даних SQL Lite;
- нереляційна база даних Redis;
- нереляційна база даних MongoDB;
- нереляційна база даних Cassandra;

- нереляційна база даних Splice;
- операційна система Ubuntu Linux 16.04;
- операційна система Debian Linux 16.04;
- операційна система Kali Linux 2018.4.

Також для розробки потрібно використати програму для тунелювання трафіку. Тунелювання являє собою захищений зв'язок між двома кінцевими точками. Для прикладу воно може використовуватися для надання доступу до локальних напрацювань в області веб-програмування. Тобто при тунелюванні локального хоста комп'ютера, користувачі, що знаходяться поза локальною мережею, також зможуть отримати доступ до протунельованого локального хоста.

В даному випадку, під час розробки, тунелювання необхідне для зручності тестування. Оскільки розробка ведеться безпосередньо на локальній машині, потрібно якимось чином створити можливість для доступу веб-хуків від Telegram Bot API до локальної машини. Саме для цього було використано тунелювання.

Для організації тунелювання було використано програмний пакет сервісу ngrok. Даний сервіс дозволяє реалізувати тунелювання локального хоста через вікно терміналу, після чого з цього ж вікна можна отримати посилання за яким буде доступний локальний хост з мережі Інтернет.

1.2 Структура системи

Основою проекту є взаємодія між викладачами та студентами. Це означає, що поставлена задача вимагає від системи розділення користувачів на різні групи, які мають різний рівень доступу до системи. Тому повинно бути реалізовано розділення, користувачів на декілька груп. Кількість груп більша ніж дві (тобто студенти та викладачі), оскільки викладачі також повинні мати певні обмеження.

Буде здійснюватися постійний односторонній зв'язок зі стороннім сервісом Telegram Bot API, який являє собою сповіщення про всі дії, що виконуються з Телеграм-ботом за допомогою веб-хуків. До найпоширеніших дій відносяться отримання приватного повідомлення, отримання повідомлення в груповому чаті та натискання вбудованих кнопок. Тому потрібно реалізувати універсальний обробник веб-хуків, який в залежності від типу дії та даних визначає наступну послідовність дій.

Також передбачається взаємодії всіх груп користувачів з файлами. Це можуть бути матеріали для навчання та інші файли, які користувачі додадуть до спільного доступу для всіх користувачів системи. Тому постає задача реалізації модуля збереження файлів, інформації про них, подальшого їх використання та передавання.

Через орієнтованість системи на великі навантаження та значну кількість користувачів, система потребує модуля для обробки користувачів. Мається на увазі слідкування за станом користувача, верифікованістю його аккаунту в системі, обробку його даних та слідкування їх зміни. Тому потрібно реалізувати даний модуль в розроблюваній системі. Також має сенс реалізувати виконання функціоналу даного модуля в фоновому режимі, задля збільшення продуктивності та швидкодії системи.

1.3 Функціональні частини системи

Студенти часто стикаються з такими питаннями як: пошук матеріалів з предмету, зв'язок з викладачами, складність пошуку їх контактних даних та інші. Також питання можуть виникати у викладачів, такі як: отримання списків студентських груп, отримання інформації про студентів, зв'язок зі студентами та інші. Завданням даного проекту є вирішення найбільш загальних та розповсюджених проблем, пов'язаних із навчальним процесом.

Загальний перелік функціоналу системи наведений в наступних пунктах даного підрозділу.

1.3.1 Реєстрація

З технічної точки зору після кожної відповіді студента дані зберігаються в відповідне поле бази даних. Після відповіді на останнє питання потрібно знайти відповідальну особу, яка повинна верифікувати нового користувача. Цей пошук відбувається частково на основі даних, які надав користувач. Також для пошуку використовується пошук за ролями, оскільки лише користувачі з певною роллю в системі є відповідальними за підтвердження даних нових користувачів. Якщо користувач вибрав роль студента, то пошук відповідальної особи відбувається за факультетом, кафедрою та студентською групою. Якщо не було надано яких-небудь даних, то пошук відбувається за останніми заповненими даними, якщо такі були зазначені.

1.3.2 Управління матеріалами

Викладачі мають змогу додати матеріали, які будуть доступні для всіх користувачів сервісу. Це можуть бути методичні вказівки або інші матеріали, корисні в навчальному процесі. В майбутньому ці матеріали можуть полегшити процес навчання як викладачам так і студентам, оскільки вони будуть в загальному доступі та в одному місці.

1.3.3 Отримання інформації про користувача

					IA52.160БАК.002 ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підп.	Дата		13

У відповідному розділі інтерактивного меню можна отримати інформацію про конкретного користувача. В інформації про викладача можна отримати додані ним матеріали.

1.3.4 Отримання інформації про групи користувачів

В розроблюваній системі, користувачі поділяються на два типи - студенти та викладачі. Студенти можуть отримати інформацію лише про свою групу в Excel файлі. У викладачів в цьому аспекті можливостей більше. В залежності від рівня доступу, викладач може отримати інформацію про списки студентських груп та списки викладачів. В отриманих списках будуть зазначені всі дані й відомі про користувача.

1.3.5 Додавання користувацьких чатів

Викладачі мають змогу автоматизувати процес додавання студентів в свої чати. Для цього потрібно створити чат, додати в нього Телеграм-бота данного сервісу та надати йому адміністраторські права. Після цього в відповідному розділі меню з'явиться розділ керування чатами. В цьому розділі можна буде побачити список доданих чатів. В кожному із чатів є можливість додати студентів обраної групи, а також видалити вже додані.

1.3.6 Адміністрування системи безпосередньо в самій системі

Підтвердження користувачів, додавання викладацьких матеріалів, створення груп та інше - всі ці дії виконуються в інтерфейсі месенджера Телеграм без потреби переходу до стороннього сервісу.

Висновки до розділу 1

В даному розділі було розглянуто варіанти як окремих частин системи так і загальну структуру побудови проекту. З огляду на поставлене завдання та можливі варіанти його реалізації, були розглянуті та обрані варіантів, такі як: веб-додаток та додаток на базі месенджера. Розглядалася можливість використання таких месенджерів WhatsApp, Viber, Телеграм (ориг. Telegram). Телеграм та його Telegram Bot API здобув велику популярність у всьому світі. Його переваги - швидкість роботи, зручність та простоту інтерфейсу, велика варіативність функціоналу тощо. Тому даний месенджер багато в чому задовольняє вимоги даного проекту.

Також були обрані сучасні та найбільш підходящі технологічні частини для використання в проекті. Були розглянуті варіанти архітектурної структури проекту та перелік функціоналу системи які в свою чергу тісно пов'язані з темою проекту. Всі варіанти відповідають вимогам до кінцевого продукту та вирішують поставлене завдання.

2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

2.1 Подібні системи взаємодії

Сфера освіти активно розвивається, це призводить до необхідності впровадження нових підходів до процесу навчання. В тому числі з розвитком інформаційних технологій можливості для оптимізації навчального процесу зросли. Це можуть бути онлайн курси, відео курси, мобільні додатки, веб-сервіси, безпосередньо віддалене навчання з використанням спеціалізованих сервісів, розроблених для цього.

Ця методологія отримала назву «електронне навчання» («e-learning»). Вона має порівняно довгу історію своєї появи, яка починається з початку ХХ століття. Для навчання використовували проекції ілюстрацій, які проектувалися за допомогою епіскопів. З плином часу і появою нових технологій, їх також почали використовувати в навчанні. Вже в середині ХХ століття в Великобританії почало зароджуватися поняття «електронного навчання». Саме там був створений перший відкритий університет, в якому була виключно заочна форма навчання. Пізніше подібні університети почали з'являтися в інших європейських країнах [1].

Дану методологію навчання часто плутають з дистанційним навчанням, мультимедійним навчанням, навчанням з використанням комп'ютерів. Основні принципи:

- відкритість освітніх ресурсів;
- децентралізація навчальної діяльності;
- використання інформаційних технологій;
- цифрові навчальні матеріали;
- відкрита ліцензія для доступу до матеріалів;
- асинхронність процесу навчання;
- розробка та надання навчальних матеріалів через цифрове освітнє середовище.

Цей підхід є перспективним і активно використовується у системі загальної освіти. Для вищої освіти доцільно проаналізувати розвиток такого педагогічного підходу як - дистанційне навчання, яке передбачає взаємодію викладача і студента на відстані і використовувалося з XVIII сторіччя, коли обмін матеріалами відбувався за допомогою пошти. Суттєві зміни відбулися з виникненням радіо і телебачення. Однак справжньої популярності дистанційне навчання набуло з появою Інтернету, коли кожен користувач мережі може створювати контент. З розвитком технологій з'явилися нова педагогічна парадигма - електронне навчання (цифрова освіта, онлайн-навчання), що безпосередньо ґрунтується на використанні інформаційно-комунікаційних технологій.

Електронне навчання можна класифікувати за способом отримання навчальних матеріалів: синхронне (вебінари, відеоконференції, спільна робота над документами) та асинхронне навчання (електронна пошта, файлообмінники, підкаст, MOOC та ін.).

У системі вищої освіти сьогодні є необхідним змішане (гібридне) навчання - освітня модель, яка поєднує цифрову та традиційну форми навчання. За умови змішаного навчання студенти суміщають роботу в аудиторіях навчального закладу з опрацюванням частини матеріалу онлайн.

Таким чином, зміна традиційних підходів до навчання, з одного боку, є цифровою революцією та стрімким розвитком інформаційно-комунікаційних технологій, а іншого, вони сприяли появі такого феномену як електронне навчання. Еволюція інформаційно-комунікаційних технологій свідчить про різноманітність технологій та інструментів, які допомагають їх розширити, і в свою чергу, це свідчить про варіативність їх залучення до навчального процесу, який також перебуває в стані радикальних змін направлених на відкритість матеріалів, більш гнучкого і зручного процесу навчання та усунення меж, таких як: географічних, культурних, вікових тощо.

Сама система передбачає самостійне навчання на основі навчальних матеріалів, до яких був наданий доступ. Перевірка закріплених знань може здійснюватися як за допомогою однакових тестів для всіх учнів, так і за допомогою індивідуальних завдань, виконання яких підтверджує отримання та засвоєння нових знань.

З точки зору розроблюваної системи, завданням методології «електронного навчання» є повна автоматизація і децентралізація процесу навчання, що йде наперекір завданню розроблюваної системи. Це обумовлено тим, що система не має на меті втручатися в навчальний процес, а повинна лише автоматизувати та спростити деякі його процеси.

2.2 Онлайн курси «Skylang» для вивчення іноземних мов

Даний сервіс являє собою онлайн школу для вивчення англійської мови. Основою школи є веб-сервіс, в якому відбувається безпосередно саме навчання. Навчальний процес являє собою набір курсів з різними завданнями. Серед можливих завдань: зіставлення слова, пропозиції до картинки, складання пропозиції з слів, аудіофайли, можливість підписати картинку, зіставлення заголовків до абзаців тексту. Всі користувачі поділяються на три категорії: учнів, викладачів та адміністраторів.

Учень проходить разом з викладачем урок і його відповіді видно викладачеві в режимі реального часу. Неправильні і правильні відповіді підсвічуються різними кольорами. Учень так само отримує можливість самостійно вести свій онлайн словник. Під час кожного уроку учень записує нові слова, які після збереження додаються в особистий словник учня, який відображається в його особистому кабінеті. Він дає студенту можливість стежити за своїм розкладом, отримати детальну інформацію про вчителя і курсах. Відслідковувати прогрес проходження занять, а також переносити

або скасовувати їх. В особистому кабінеті так само виведена інформація про оплату курсів.

Адміністратор в даній системі також виконує роль менеджера, оскільки займається оформленням користувачів, записом їх для тестувань та оформленням на перше пробне заняття (Рисунок 2.1).

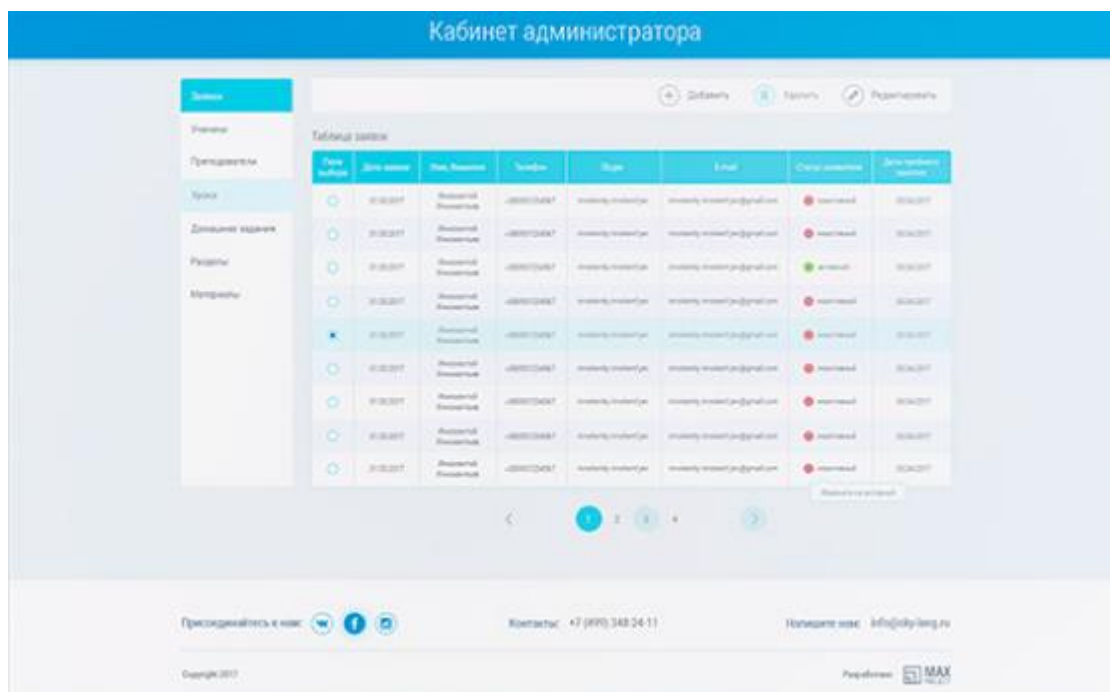


Рисунок 2.1 - Приклад кабінета адміністратора

Взаємозв'язок учнів та викладачів може відбуватися в два способи. Перший, це зв'язок всередині системи за допомогою особистих повідомлень. Також за допомогою повідомлень учні можуть мати зв'язок з адміністраторами сервісу. Під час проведення онлайн уроку вчитель може бачити всіх учнів, які перебувають «на занятті», а також в режимі реального часу може бачити, які відповіді на питання дали учні та їх правильність. Другим способом взаємозв'язку між учнями та викладачами є сторонній сервіс Skype. Під час реєстрації, учні вказують своє ім'я в цьому сервісі задля подальшого використання. З його допомогою учні можуть отримати

особисту консультацію у викладача, а також, за потреби, можуть дати свої настанови учневі.

Загалом цей сервіс схожий на розроблювану систему, з тими лише відмінностями, що він виконаний як веб-сервіс, а також він є невід’ємною та головною частиною процесу навчання.

2.3 Веб-сервіс з онлайн курсами різної направленості «L-a-b-a»

Ця освітня платформа була створена відносно недавно. Основним направленням сервісу є швидке та ефективне навчання. Через це вона являє собою веб-сервіс, в якому є як учні, так і викладачі. Сам процес навчання повністю направлений на самоосвіту.

В даному сервісі учні записуються на певні курси, яких в системі дуже багато і всі вони розділені на блоки, в залежності від категорії. Всього серед курсів налічується близько п’ятнадцяти категорій, наприклад: для особистого росту, енергетика, бізнес, продажі, бізнес-аналітика, курси дизайну, курси направлені на освоєння офісного програмного забезпечення та інші. Після запису на курс, учні отримують доступ до відеоматеріалів, які містять записи вебінарів, на яких розглядаються важливі питання стосовно тематики курсу. Учні мають змогу зберегти всі відеоматеріали, задля подальшого використання, а саме повторних переглядів для повторення пройденого матеріалу.

Взаємодія викладачів та студентів в даному сервісі відбувається за допомогою стороннього сервісу, а саме месенжера Телеграм. Саме там в учнів є можливість поставити свої запитання до викладача курсу. Процес пошуку викладачів та спілкування не автоматизований. Учні самі знаходять потрібних їм викладачів та потрібні діалоги. Для зручності пошуку використовуються прямі посилання для вступу до групових діалогів.

Сервіс є вдалим вирішення проблеми гнучкого та ефективного процесу навчання, але не вирішує основної задачі поставленої перед розроблюваною системою.

2.4 Електронний кампус

Електронний кампус являє собою веб-сервіс для організації та спрощення навчального процесу в Національному технічному університеті України «Київському політехнічному інституті імені Ігоря Сікорського». Ця система надає централізований доступ до всіх навчальних та методичних матеріалів. Також вона надає можливість спілкування між викладачами та студентами. В відповідному розділі можна дізнатися інформацію про свого куратора та його контактні дані. Сам сервіс створений досить давно, на сьогоднішній день його можна вважати застарілим, і зараз він переживає етап змін, як в графічному інтерфейсі, так і у функціоналі.

В цілому завдання даної системи є найбільш наближеним до завдання поставленого перед розроблюваною системою.

2.5 Телеграм-бот «Предложка лайва»

Телеграм-бот «Предложка лайва» створений за допомогою іншого бота, а саме «Livegram Bot», який призначений для створення ботів зворотнього зв'язку, яким є розглянутий Телеграм-бот. Його основною ідеєю є спрощення процесу адміністрування Телеграм-каналів та реалізації зворотного зв'язку з його підписниками. Телеграм-канал являє собою спільноту, в якій адміністратори створюють публікації, а всі підписники каналу можуть їх переглядати. Принцип роботи бота доволі простий, всі надіслані йому повідомлення пересилаються власнику бота, після чого

власник може відповісти на ці повідомлення. До переваг можна віднести відсутність потреби розробки якого-небудь функціоналу та мінімальну кількість налаштувань. Що правда, як наслідок, такий бот має низьку функціональність. Загалом завдання бота майже не співпадає із завданням розроблюваного проекту та використовується лише для вирішення вузького спектру задач.

2.6 Телеграм-бот «ЦФВС КПІ ім. Ігоря Сікорського»

Цей Телеграм-бот виконує роль довідника. В боті реалізоване меню для зручного керування та навігації. Він дає змогу дізнатися про секції факультету фізичного виховання, інформацію, потрібну для реєстрації в секціях, посилання на форми реєстрації на секції та інше. Меню реалізоване за допомогою груп клавіш, які з'являються замість звичайної клавіатури. Таке меню доволі просте в реалізації та використанні, воно цілком задовольняє функціональні вимоги даного сервісу.

Загалом дана система має простий та зрозумілий інтерфейс. Стійкість системи забезпечена не гнучкою чіткою реалізацією, яка оброблює лише певні запити. Сервіс лише частково переслідує завдання поставлене перед розроблюваною системою.

2.7 Телеграм-бот «KPI schedule»

Телеграм-бот «KPI schedule» створений для контролю розкладу занять та сесії. Не дивлячись на це, бот має широкий функціонал пов'язаний з розкладом. За допомогою цього бота можна дізнатися розклад на поточний та наступний тиждень навчання, в якому будуть лише назви предметів та номер пари, на якій відбудеться заняття. Також можна отримати повний розклад на два тижні, в якому також будуть зазначені

імена викладачів, які проводять пару. Є можливість увімкнути автоматичне сповіщення про розклад на наступний день. Ці сповіщення можна налаштувати в самому ботові, після чого студенту напередодні буде надсилатися розклад у визначений ним час. Загалом даний бот є інформативним і лише частково переслідує мету поставлену перед проектом.

					ІА52.160БАК.002 ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підп.	Дата		23

Висновки до розділу 2

У розділі розглянуті аналоги, які в різній мірі підходять до вимог, поставлених перед проектом. Не дивлячись на те, що деякі з них використовують ті ж самі технології та сервіси, завдання поставлені перед аналогами відмінні від тих, які повинен вирішувати розроблюваний проект.

А саме, було розглянуто поняття «електронного навчання», яке має на увазі повну автоматизацію і децентралізацію процесу навчання, що йде наперекір завданню розроблюваної системи. Також онлайн курси «Skylang» та «L-a-b-a», електронний кампус, телеграм-боти: «Предложка лайва», «Центр фізичного виховання та спорту КПП ім. Ігоря Сікорського», «KPI schedule», це все частково переслідує завдання чи мету, які поставлені перед проектом.

Отже, можна зробити висновок, що серед існуючих рішень не було знайдено такого, що в повній мірі переслідує ті самі цілі, які поставлені перед даним проектом.

3 РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1 Опис розробленого функціоналу

Реалізація дипломного проекту виконана в якості веб-додатку, який не має Front End частини, тобто візуального інтерфейсу, а лише Back End частину, тобто серверну. В якості візуального інтерфейсу використовується месенджер Телеграм, який надає змогу взаємодіяти з користувачами за допомогою віртуального користувача, який являє собою бота, що керується розроблюваною системою.

Відсутність потреби розробки графічного інтерфейсу обумовлена завданням дипломного проекту. Оскільки розробка сервісу ведеться задля спрощення та пришвидшення деяких навчальних процесів. Графічний інтерфейс створюватиме певні труднощі, які пов'язані з необхідністю переходу з основного інтерфейсу сервісу, тобто месенджера Телеграм, до іншого веб-сервісу.

В наступних пунктах даного підрозділу описаний функціонал, який був виконаний в дипломному проекті.

3.1.1 Реєстрація

Реєстрація влаштована незвично з точки зору існуючих систем. Вона являє собою запит на підтвердження користувача в системі. Спершу користувачеві, який має намір зареєструватися в системі потрібно знайти певного Телеграм-бота в месенджері Телеграм. Надалі всі дії пов'язані з розроблюваною системою будуть відбуватися через цього Телеграм-бота. Після знаходження бота потрібно його запустити відповідною кнопкою «Start». Після цієї дії користувача, за допомогою вебхуку, буде відправлене відповідне сповіщення на сервер, в якому описано що відбулося і який користувач в цьому був задіяний. Після його отримання на сервері, цей

					ІА52.160БАК.002 ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підп.	Дата		25

запит буде оброблений певними перевірками. Серед них - перевірки користувача та його активності.

Перевірка користувача являє собою пошук за унікальним ідентифікатором і якщо користувач знайдений в системі та його можна вважати зареєстрованим (за певною ознакою), то процес реєстрації пропускається. В іншому випадку (якщо користувач невідомий для системи, або його статус свідчить про незавершену реєстрацію) ініціюється процес перевірки активності. Перевірка активності - це перевірка стану процесу реєстрації, вона виконується в декілька етапів. Оскільки це був перший етап, на сервері згенерується повідомлення про потребу відправки свого контакту ботові, після чого воно відправиться користувачеві.

Потім користувач повинен відправити ботові свій контакт за допомогою відповідної кнопки. Контакт відправиться на сервер за допомогою вебхуку. Будуть здійснені перевірка та збереження контактної інформації, після чого надішлеться наступне повідомлення з проханням вказати в якості кого він реєструється (студент чи викладач). Далі користувач отримає повідомлення із запитанням студент він чи викладач. Також до цього повідомлення прикріплено дві кнопки за допомогою яких потрібно відповісти. Всі відповіді користувача будуть збережені на сервері.

Наступні кроки відбуваються за тим самим алгоритмом. Користувач обирає свій факультет та спеціальність. Якщо користувач виступає в ролі студента, йому потрібно буде обрати студентську групу, в якій він навчається. Для викладача цей крок пропускається.

Далі йде ряд запитань, до яких немає кнопок з готовими відповідями, їх потрібно написати самостійно та відправити. Потрібно по черзі назвати свої прізвище, ім'я, по батькові, контактний email та вказати причину реєстрації, також іншу інформацію за бажанням користувача.

Після проходження цих етапів, всі зібрані дані надсилаються відповідальному за підтвердження користувачеві. Якщо все правильно, після підтвердження, новий користувач зможе використовувати систему.

3.1.2 Інтерактивне меню

Інтерактивне меню являє собою повідомлення, під яким розміщені кнопки. Кнопки є такими ж, як і під час реєстрації, але за допомогою них можна переходити до інших розділів меню. В повідомленні написана назва розділу меню, в якому на даний момент знаходиться користувач. Саме меню складається з кнопок, на яких написані назви розділів, до яких перейде користувач натиснувши на них. Меню може бути викликане користувачем за допомогою спеціальної команди «/menu», яку потрібно надіслати ботові.

3.1.3 Додавання користувацьких чатів

Викладачі мають змогу автоматизувати процес додавання студентів в свої чати. Для цього потрібно створити чат, додати в нього Телеграм-бота данного сервісу, та надати йому адміністраторські права. В відповідному розділі меню з'явиться розділ керування чатами. В цьому розділі можна буде побачити список доданих чатів. В кожному із чатів є можливість додати студентів обраної групи, а також видалити вже додані. Після додавання Телеграм-бота до чату, він отримує сповіщення про це, а також інформацію про користувача який додав бота та про чат до якого він був доданий. Після цього ініціюється діалог з користувачем для уточнення призначення новоствореного чату.

3.1.4 Автоматизовані опитування

Цей функціонал призначений для отримання інформації про користувачів. Оскільки кількість такої інформації може бути дуже великою, є сенс отримувати її порціями для зручності користувача. На основі цього функціоналу працює також реєстрації.

3.2 Опис запитів до Telegram Bot API

Згідно з офіційною документацією Telegram Bot API, всі запити до сервісу повинні виконуватись через протокол HTTPS, який являє собою розширення протоколу HTTP, за допомогою шифрування, з ціллю підвищення безпеки.

API підтримує GET та POST методи протоколу HTTP. У відповідь, сервіс надсилає об'єкт JSON. В об'єкті може міститися як позитивна, так і негативна відповідь сервісу. Негативна відповідь - це код помилки та її короткий опис. В свою чергу, у позитивній відповіді буде інформація про виконаний запит. В наступних пунктах даного підрозділу описані всі запити, які були використані в проекті [2].

3.2.1 Типи зворотного зв'язку

Сутності оновлень, які можуть бути повернуті:

- 1) message - нове повідомлення, яке було надіслано до групи, супергрупи або особистого чату з ботом. Містить інформацію про відправника, отримувача та сам вміст повідомлення;
- 2) edited_message - сповіщення про зміну вмісту повідомлення;

Зм.	Аркуш	№ докум.	Підп.	Дата

- 3) `channel_post` - новий пост, який був відправлений до каналу, на який підписаний бот. Містить інформацію про відправника та сам вміст поста;
- 4) `edited_channel_post` - сповіщення про зміну вмісту поста на каналі;
- 5) `inline_query` - новий вхідний вбудований режим;
- 6) `chosen_inline_result` - результат нового вбудованого режиму, який вибрав користувач;
- 7) `callback_query` - зворотній зв'язок вбудованого режиму;
- 8) `poll` - опитування, яке являє собою анонімне голосування. В об'єкті даного сповіщення надходять відомості про зміну стану опитування. Бот має доступ лише до закритих опитувань та до створених безпосередньо ботом.

3.2.2 Метод «getUpdates»

Метод «getUpdates» використовується для отримання відомостей про нові події які відбулися з ботом. Таким чином можна дізнатися про повідомлення, які надсилаються ботові або про те, що хтось натиснув на кнопку створену ботом. Після виконання запиту, у відповідь надсилається масив з діями, які були виконані з ботом.

Даний метод приймає чотири необов'язкові параметри:

- 1) `offset` - зміщення, починаючи з якого будуть надіслані оновлення. Даний параметр приймає ідентифікатор оновлення;
- 2) `limit` - даний параметр задає кількість оновлень, які будуть отримані за один запит. Приймаються значення від 1 до 100. Значення за замовчуванням 100;
- 3) `timeout` - час, заданий в секундах, протягом якого будуть проводитися опитування API на рахунок нових оновлень. Значення за замовчуванням 0;

4) allowed_updates - типи оновлень, які будуть зазначені у відповіді.

Метод використовує «довге опитування» (англ. long polling), це різновид push-сповіщень. Технологія довгого опитування є не справжнім push-сповіщенням, а лише його імітацією. Це обумовлено тим, що «довгі опитування» ініціюються стороною приймача, і за природою є схожими на звичайні HTTP-запити, з тою лише відмінністю, що запит відбувається довше. Загалом технологія «довгого опитування» використовується в тих рідких випадках, коли немає можливості увімкнути push-сповіщення. Також вона може бути застосована для тестування.

3.2.3 Метод «setWebhook»

Метод використовується для налаштування push-сповіщень від API про нові оновлення. Після налаштування всі нові сповіщення будуть надсилатися на заздалегідь визначену адресу, безпосередньо після дії, яка описана в оновленні. Вміст push-сповіщення містить інформацію лише про одне оновлення, на відміну від методу «getUpdates», оскільки для кожного оновлення ініціюється індивідуальне push-сповіщення.

Даний метод може приймати такі параметри:

- 1) url - посилання, на яке будуть відправлятися сповіщення;
- 2) certificate - файл сертифікату із ключем доступу, якщо він використовується при зверненні до серверу;
- 3) max_connections - кількість максимальних підключень одночасно, приймаються значення від 1 до 100. За замовчуванням 40;
- 4) allowed_updates - типи оновлень, які будуть зазначені у відповіді.

Push-сповіщення мають деякі умови використання, через які, іноді, неможливо їх використовувати. В першу чергу домен, до якого вони будуть надходити, повинен бути доступним по захищеному протоколу HTTPS. Якщо немає можливості налаштувати сертифікат для використання HTTPS

протокола, потрібно використовувати метод «getUpdates» описаний у попередньому пункті 3.3.1. Слід зауважити, що одночасне використання push-сповіщень та методу «getUpdates» неможливе, через обмеження самого API.

3.2.4 Метод «getWebhookInfo»

Метод для отримання інформації про налаштування push-сповіщень. Даний метод не має параметрів. У відповідь надсилаються такі інформаційні поля:

- 1) url - адреса, за якою надсилаються сповіщення про оновлення;
- 2) has_custom_certificate - параметр, що вказує про використання сертифікату для доступу;
- 3) pending_update_count - кількість оновлень, які ще не були отримані;
- 4) last_error_date - дата останньої помилки;
- 5) last_error_message - опис останньої помилки;
- 6) max_connections - кількість максимальних підключень;
- 7) allowed_updates - типи оновлень, які будуть зазначені у відповіді.

3.2.5 Метод «sendMessage»

Основний метод для відправки повідомлень. З його допомогою можна надіслати звичайне текстове повідомлення. Крім цього, за допомогою методу «sendMessage» до повідомлення можна прикріпити Inline Keyboard Markup та Reply Keyboard Markup, які являють собою кнопки, закріплені за певним повідомленням. Після використання кнопки, генерується push-сповіщення про подію натискання на кнопку. Після цього буде надіслане сповіщення з типом «callback_query».

Метод приймає такі параметри:

					ІА52.160БАК.002 ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підп.	Дата		31

- 1) chat_id - унікальний ідентифікатор чату, в який буде надіслано повідомлення;
- 2) text - текст повідомлення;
- 3) disable_notification - параметр приймає булеве значення, правда або неправда. Якщо правда, сповіщення про це повідомлення буде заглушене у користувачів;
- 4) reply_to_message_id - унікальний ідентифікатор повідомлення, відповіддю до якого є надіслане повідомлення;
- 5) reply_markup - об'єкт вбудованих кнопок, який описує їх тип, порядок та дію, яка буде відбуватися при їх натисканні.

3.2.6 Метод «sendContact»

Даний метод є варіацією методу «sendMessage», який описаний в пункті 3.3.5. Метод «sendContact» використовується для відправки контакту користувача. Для його використання потрібно знати деякі особисті дані користувача. В наступному переліку описані параметри, які може приймати метод з вказанням обов'язкових:

- 1) chat_id - унікальний ідентифікатор чату, в який буде надіслано повідомлення. Обов'язковий параметр;
- 2) phone_number - номер телефону користувача. Обов'язковий параметр;
- 3) first_name - ім'я користувача, вказане в контакті. Обов'язковий параметр;
- 4) last_name - прізвище користувача, вказане в контакті;
- 5) disable_notification - параметр приймає булеве значення, правда або неправда. Якщо правда, сповіщення про це повідомлення буде заглушене у користувачів;

- 6) `reply_to_message_id` - унікальний ідентифікатор повідомлення, відповіддю до якого є надіслане повідомлення;
- 7) `reply_markup` - об'єкт вбудованих кнопок, який описує їх тип, порядок та дію, яка буде відбуватися при їх натисканні.

3.2.7 Метод «`sendChatAction`»

Метод, призначений для сповіщення користувача про дії, які виконує бот. Наприклад, при переписці в месенджері між двома користувачами, коли один з користувачів почне писати повідомлення, у іншого в відповідному місці з'явиться сповіщення про те, що робить його співбесідник, наприклад, пише повідомлення або завантажує фотографію. Таке саме сповіщення про свої дії може надсилати бот.

Параметри, які приймає даний метод:

- 1) `chat_id` - унікальний ідентифікатор чату, в який буде надіслано повідомлення. Обов'язковий параметр;
- 2) `action` - назва дії, про яку буде проінформовано в чаті. Обов'язковий параметр.

3.2.8 Метод «`sendPhoto`»

Метод для надсилання фотографій. Він є інтерпретацією методу «`sendMessage`», який описаний в пункті 3.3.5, з тою лише відмінністю, що даний метод використовується для надсилання фотографій, а не лише текстової інформації.

Метод приймає наступні параметри:

- 1) `chat_id` - унікальний ідентифікатор чату, в який буде надіслано повідомлення. Обов'язковий параметр;

- 2) photo - параметр приймає рядок, який повинен бути унікальним ідентифікатором існуючої фотографії або сам файл фотографії. Обов'язковий параметр;
- 3) caption - підпис до фотографії;
- 4) disable_notification - параметр приймає булеве значення, правда або неправда. Якщо правда, сповіщення про це повідомлення буде заглушене у користувачів;
- 5) reply_to_message_id - унікальний ідентифікатор повідомлення, відповіддю до якого є надіслане повідомлення;
- 6) reply_markup - об'єкт вбудованих кнопок, який описує їх тип, порядок та дію, яка буде відбуватися при їх натисканні.

3.2.9 Метод «getUserProfilePhotos»

Метод використовується для отримання унікальних ідентифікаторів фотографій профілю певного користувача. При використанні методу можна задати кількість фотографій, про які будуть повернуті відомості та зміщення відносно першої фотографії, якщо нам не потрібні перші фотографії.

Метод приймає наступні параметри:

- 1) user_id - унікальний ідентифікатор користувача. Обов'язковий параметр;
- 2) offset - зміщення. Задає порядковий номер фотографії, яка буде першою;
- 3) limit - кількість фотографій, про які буде надіслана інформація.

3.2.10 Метод «getFile»

Метод для отримання файлу за його унікальним ідентифікатором. Метод «getUserProfilePhotos», описаний в пункті 3.3.9 повертає ідентифікатори фотографій користувача, а для отримання самих файлів фотографій використовується даний метод. Після отримання файлу, його можна зберегти на сервері, якщо в цьому є потреба.

Метод приймає параметр file_id - унікальний ідентифікатор файла.

3.2.11 Метод «exportChatInviteLink»

Метод для створення посилання-запрошення для чату. Після використання методу, для чату створюється нове посилання-запрошення (замінюючи старе, якщо воно існувало), яке буде доступне для бота в методі «getChat», описаному в пункті 3.3.13. Для створення посилання-запрошення, бот повинен бути учасником чату та мати адміністраторські права. Після створення посилання-запрошення, бот може додавати всіх бажаючих користувачів до чатів, коли в цьому є потреба.

Метод приймає параметр chat_id - унікальний ідентифікатор чата. Обов'язковий параметр.

3.2.12 Метод «kickChatMember»

Метод для видалення та блокування користувачів в чатах. З його допомогою можна заблокувати користувача на певний, визначений заздалегідь час. Якщо не вказувати час, на який буде заблокований користувач, він буде видалений з чату назавжди.

Метод приймає наступні параметри:

- 1) chat_id - унікальний ідентифікатор чату, в якому буде видалений або тимчасово заблокований користувач. Обов'язковий параметр;

- 2) user_id - унікальний ідентифікатор користувача, який буде видалений або тимчасово заблокований в чаті. Обов'язковий параметр;
- 3) until_date - час, на який буде заблокований користувач. Якщо даний параметр не вказаний або некоректний, користувач буде видалений з чату повністю.

3.2.13 Метод «getChat»

Метод для отримання інформації про чати. У відповідь надсилається така інформація, як: назва чату, його опис, посилання-запрошення, якщо воно було згенероване, інформація про фотографії чату.

Метод приймає параметр chat_id - унікальний ідентифікатор чату, інформація про який буде надіслана у відповідь.

3.2.14 Метод «deleteMessage»

Метод для видалення повідомлень, має обмеження, які ускладнюють використання методу:

- повідомлення повинно бути надісланим менше ніж за 48 годин до моменту видалення;
- для видалення своїх, а також повідомлень інших користувачів в чатах, які не є приватними, бот повинен мати певні адміністраторські дозволи.

Через такі обмеження, метод не є достатньо ефективним, оскільки вимагає складної логіки слідкування за повідомленнями, яка дасть змогу розуміти, які з повідомлень можна видалити, а які не можна.

Метод приймає наступні параметри:

- 1) chat_id - унікальний ідентифікатор чату, в якому знаходиться повідомлення для видалення. Обов'язковий параметр;

- 2) `message_id` - унікальний ідентифікатор повідомлення, яке буде видалене. Обов'язковий параметр.

3.2.15 Метод «`editMessageCaption`»

Метод для зміни опису повідомлень, якими були надіслані певні медіа ресурси. Це можуть бути описи фотографій, документів, аудіофайлів та відеофайлів. Також за допомогою даного методу можна змінити, додати, або видалити вбудовані в повідомлення кнопки зворотного виклику.

Метод приймає наступні параметри:

- 1) `chat_id` - унікальний ідентифікатор чату, в якому знаходиться повідомлення, яке буде відредаговане. Є обов'язковим параметром, якщо не вказано `inline_message_id`;
- 2) `message_id` - унікальний ідентифікатор повідомлення, яке буде відредаговане. Є обов'язковим параметром, якщо не вказано `inline_message_id`;
- 3) `inline_message_id` - унікальний ідентифікатор вбудованого повідомлення. Є обов'язковим параметром, якщо не вказані `chat_id` та `message_id`;
- 4) `caption` - новий опис повідомлення;
- 5) `reply_markup` - об'єкт вбудованих кнопок, який описує їх тип, порядок та дію, яка буде відбуватися при їх натисканні.

3.2.16 Метод «`editMessageText`»

Метод для редагування текстових повідомлень. З його допомогою можна змінити текст повідомлення, а також змінити, додати, або видалити вбудовані в повідомлення кнопки зворотного виклику.

Метод приймає наступні параметри:

					IA52.160БАК.002 ПЗ	Аркуш 37
Зм.	Аркуш	№ докум.	Підп.	Дата		

- 1) chat_id - унікальний ідентифікатор чату, повідомлення якого буде відреаговане. Є обов'язковим параметром, якщо не вказано inline_message_id;
- 2) message_id - унікальний ідентифікатор повідомлення, вміст якого буде відреагований. Є обов'язковим параметром, якщо не вказано inline_message_id;
- 3) inline_message_id - унікальний ідентифікатор вбудованого повідомлення. Є обов'язковим параметром, якщо не вказані chat_id та message_id;
- 4) text - новий текст повідомлення;
- 5) reply_markup - об'єкт вбудованих кнопок, який описує їх тип, порядок та дію, яка буде відбуватися при їх натисканні.

Практична частина дипломного проекту виконана в якості веб-додатку, що має лише Back End частину з візуальним інтерфейсом у месенджері Телеграм. У розділі було описано функціонал, який використаний у веб-додатку, а саме: реєстрація, інтерактивне меню, додавання користувацьких чатів, автоматизовані опитування для отримання інформації про користувача, опис запитів тощо, а також деякі методи взаємодії з ботом.

В ході розробки було виявлено складність взаємодії сервісу з Telegram Bot API, пов'язану з вимогами до налаштування вебхуків. Складність полягає у тому, що вебхуки можуть бути підключені лише по захищеному посиланню. Це спонукало до використання програмного пакету для тунелювання. Також це дало змогу вести розробку проекту з можливістю локально тестувати функціонал.

Зм.	Аркуш	№ докум.	Підп.	Дата

4 АРХІТЕКТУРА СЕРВЕРНОЇ ЧАСТИНИ

Зазвичай сервер являє собою віддалену робочу машину, яка не має графічного інтерфейсу. Всі його потужності задіяні для роботи одного або декількох проектів. На сервері встановлені лише ті програмні пакети, яких потребує проект.

4.1 Технології використані при розробці проекту

При вдалому поєднанні технологій та їх налаштувань можна досягти високої функціональності та швидкодії сервісу, тоді як невдале поєднання може призвести до некоректної роботи та навіть непрацездатності системи. Тому це дуже важливий етап розробки. В наступних пунктах даного підрозділу описані обрані технології для розробки даного проекту.

4.1.1 Веб-сервер Apache2

В якості веб-сервера було обрано Apache2, оскільки він виправдовує очікування щодо функціоналу, який він надає. Веб-сервер зарекомендував себе, як проста та швидка технологія. Ця технологія має велику кількість налаштувань, що пов'язано з її функціональністю. Процес установки та налаштування детально описаний на офіційному сайті [3]. Його використання є вільним, без ліцензій та підписок. Взаємодія з веб-сервером, тобто налаштування та запуск виконується через командний рядок за допомогою системних команд [4]. Unix-подібна операційна система має переваги, оскільки майже всі дії в ній можна виконати через командний рядок.

4.1.2 Мова програмування PHP 7.2

PHP - мова програмування, яка обрана через велику розповсюдженість та широку підтримку, що дає змогу знаходити відповіді на самі часті та розповсюджені запитання. Хоча на сьогоднішній день найактуальніша версія мови це 7.3, але вона не була обрана для розробки оскільки її альфа-версія була випущена в жовтні 2018 року, що означає що на момент початку розробки, ця версія не мала стабільної версії.

PHP - це серверна мова створення сценаріїв (або сторони сервера), розроблена спеціально для Web. У HTML-сторінку можна впровадити код PHP, який буде виконуватися при кожному її відвідуванні. Код PHP інтерпретується Web-сервером і генерує HTML або інший висновок, що спостерігається відвідувачем сторінки [5]. Загальне влаштування процесів мови має вигляд «запит - відповідь» (Рисунок 4.1). В реальності відбувається більш складний запит, послідовність якого відображає Рисунок 4.2.

Через це дану мову іноді називають «мертвою мовою». Це зовсім не пов'язано зі спадом її популярності в останні роки, процес запускається, виконується, а потім «помирає», тобто закінчує виконання шляхом закінчення скрипта, який виконувався [6].

Попередньою версією перед 7.0 була версія 5.6. Це обумовлено тим, що під час розробки 6 версії були внесені глобальні зміни структуру та функціонал, які виявилися незворотними, а також не виправданими, тому шоста версія була пропущена, а наступною новою версією стала сьома. Випуск цієї версії став грандіозним проривом для PHP, оскільки, в першу чергу, швидкодія в порівнянні з попередньою версією зросла майже вдвічі.

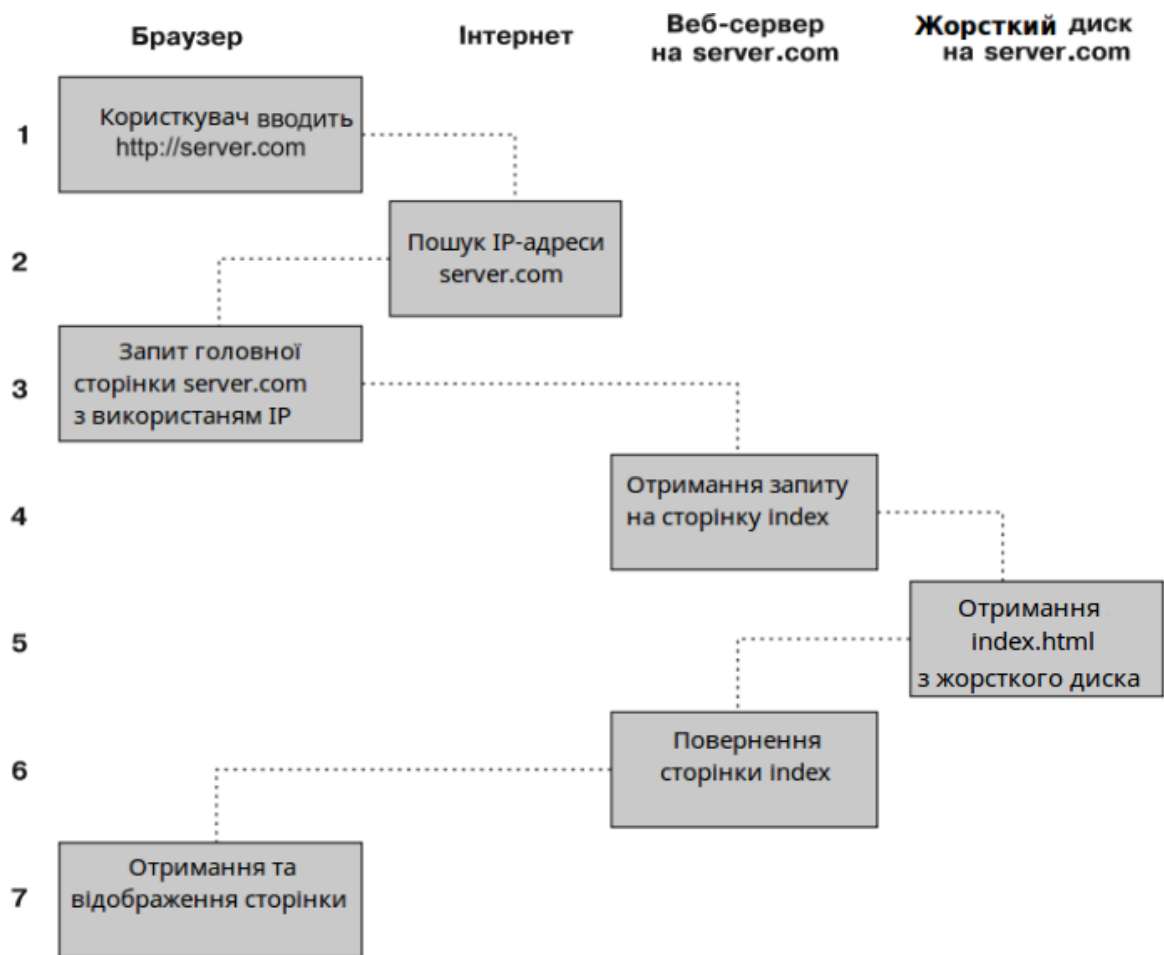


Рисунок 4.1 - Основна послідовність процесу «запит - відповідь» між клієнтом і сервером

Також були позитивні зміни в самій мові, а саме: більш компактний код, з'явилися нові логічні оператори, можливість явно прописувати типи змінних та типи даних, які повертаються, була покращена логіка обробки помилок, зросла функціональність та гнучкість мови [7]. Загалом у PHP 7 був прибрааний застарілий код, а також було дано зелене світло новим можливостям і майбутнім поліпшенням в області ефективності.

Незважаючи на часткову втрату зворотної сумісності з попередніми версіями, більшість виникаючих проблем легко вирішувати.

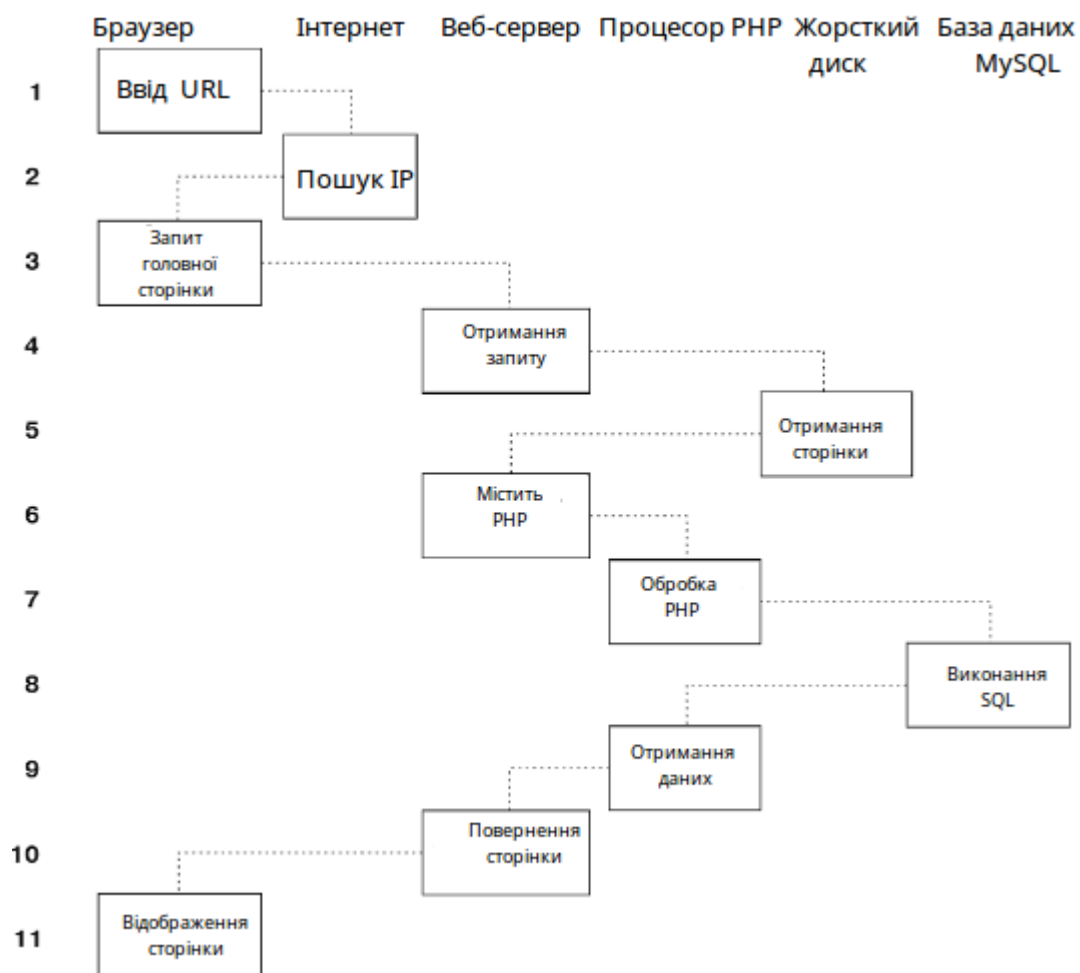


Рисунок 4.2 - Модель запиту із інтерпретацією PHP

В наступних версіях (7.1, 7.2, 7.3) вже не було такого стрімкого зростання швидкодії, але з кожною новою версією зростала функціональність мови.

4.1.3 Фреймворк PHP Laravel 5.7

Даний фреймворк був обраний для розробки через переваги над іншими фреймворками даної мови. Одним із основних факторів є простота. Він зарекомендував себе як фреймворк з «низьким порогом входу». Це означає що новачкам в даній галузі або розробникам, які змінили основну мову програмування на PHP, набагато простіше буде розібратися та

навчитися повноцінно використовувати цей фреймворк, ніж інші. Він не єдиний серед тих, що мають «низький поріг входу» [8]. Але більшість аналогів є неповноцінними і являють собою мікрофреймворки, як наприклад Laravel Lumen. На відміну від них Laravel, в першу чергу, є повноцінним фреймворком з великою кількістю базово встановлених та доступних для встановлення пакетів.

В своєму «арсеналі», даний фреймворк має такий вбудований функціонал, як: черги та можливість гнучкої роботи з ними, розширення для утиліти Unix-подібних систем, яка дозволяє використовувати команди в заданий час (cron від грец. Chronos - час), вбудовану ORM Eloquent (англ. Object-relational mapping, Об'єктно-реляційна проекція), яка дає змогу зручної взаємодії з реляційними базами даних [9]. При вдалому використанні за допомогою цього функціоналу можливо створити гнучку, швидку та потужну систему.

4.1.4 Реляційна база даних MySQL

Дана технологія являє собою багатопоточний сервер баз даних. В цілому характеризується простотою, стійкістю та швидкодією. Проявляється з найкращого боку в Unix-подібних операційних системах, оскільки в них є підтримка багатопоточності, що дуже позитивно відбивається на продуктивності системи. Система підходить для малих та середніх за масштабністю систем. Реляційні бази даних на сьогоднішній день є, мабуть, найбільш часто використовуваними. Вони засновані на потужному теоретичному базисі реляційної алгебри. Вони побудовані на основі відносин, зазвичай званих таблицями. Таблиця являє собою саме те, що і повинна - таблицю з даними. Якщо ви коли-небудь мали справу з електронною великоформатною таблицею, вважайте, що досвід роботи з реляційними таблицями у вас є. Кожен стовпець у таблиці має унікальне

ім'я і містить різну інформацію. Кожному стовпцю відповідає певний тип даних. Стовпці іноді називають також полями або атрибутами [5]. Довгий час база даних програвала за характеристиками такому конкуренту як PostgreSQL. Конкурентна база даних мала більш високий «поріг входу» ніж MySQL, але в свою чергу давала більш стійку та швидку систему для збереження даних. Вона витримувала більші навантаження ніж обрана в проекті база даних, через що користувалася більшою популярністю. Але з виходом MySQL 5.5 було внесено багато змін, які вплинули на швидкодію, масштабованість та цілісність бази загалом, що дало змогу обраній базі конкурувати з PostgreSQL, при цьому маючи нижчий «поріг входу». У випадках великої кількості даних великого значення набуває використання реляційних баз даних з структурованою системою запитів, оскільки сама по собі мова SQL є структурованою мовою запитів [10]. І MySQL, будучи абсолютно безкоштовною і встановленою на величезній кількості веб-серверів системою, являє собою надійну і виключно швидкодіючу систему управління базами даних, що використовує команди, схожі на прості англійські слова. Вищим рівнем структури MySQL є база даних, всередині якої можна мати одну або кілька таблиць, що містять ваші дані.

4.1.5 Веб-додаток phpMyAdmin

Даний веб-додаток являє собою інтерфейс для адміністрування реляційної бази даних MySQL. Він надає зручний інтерфейс для роботи з даними баз даних, їх таблицями, привілегіями на доступ до баз даних. З його допомогою можна імпортувати та експортувати як окремі таблиці, так і бази даних повністю. В спеціальному командному рядку можна використовувати мову SQL для роботи з базами даних. Dodatok надає функціонал, який можна виконати через термінал або SQL запитами, але в

зручному графічному інтерфейсі, який пришвидшує та спрощує роботу з реляційною базою даних MySQL.

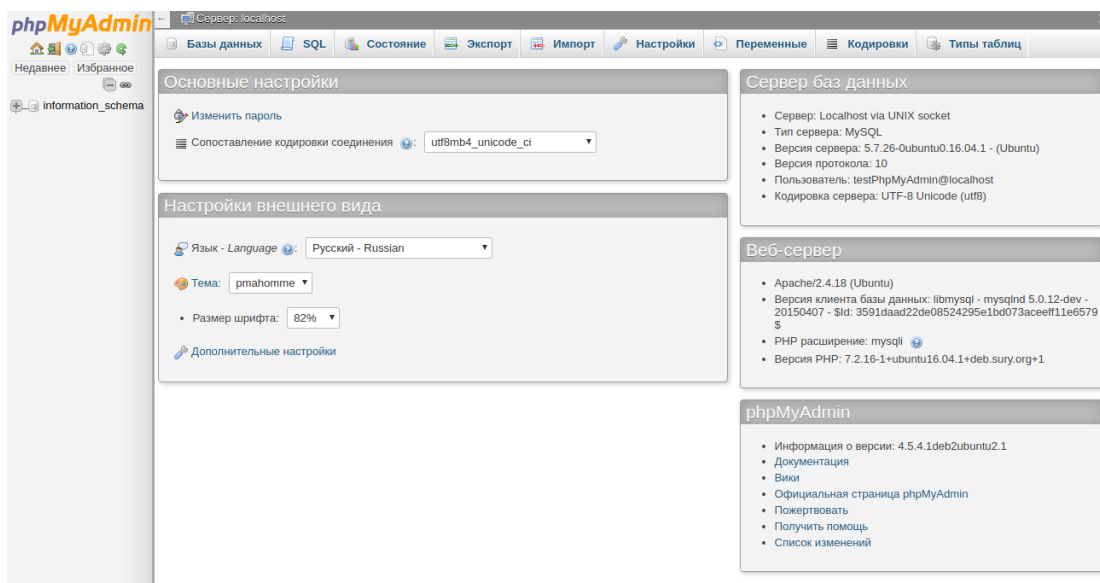


Рисунок 4.3 - Графічний інтерфейс веб-додатку phpMyAdmin

4.1.6 Нереляційна база даних Redis

Нереляційні бази даних або NoSQL (від англ. not only SQL - не тільки SQL) являють собою ряд підходів до вирішення проблеми масштабованості та доступності за рахунок атомарності та узгодженості даних. Під атомарністю слід розуміти операцію, яка не може бути виконана частково. Такі операції є ресурсами, яким надається доступ до потоку для виконання або блокується, якщо потік зайнятий, до моменту звільнення потоку [11]. Загальна модель нереляційних баз даних є відмінною від класичних реляційних систем управління базами даних з доступом до даних ресурсами мови SQL. Часто в проектах використовують одразу кілька баз даних, одну реляційну, а іншу нереляційну. Вибір конфігурації проекту завжди залежить від завдання та поставленої мети. Нереляційна база даних може стати рішенням для нескладних, але частих задач, які потребують малих

затрат ресурсів та високої швидкодії. Для таких задач погано підійде реляційна база даних, оскільки найчастіше вона виступає в ролі основної бази даних в проекті, а також часті короткі звернення до неї можуть витратити невиправдано велику кількість ресурсів [12]. Тому в такому випадку часто використовують нереляційні бази даних.

Процес установки та налаштування детально описаний на офіційному сайті [13]. Універсальність Redis і широкий вибір варіантів використання дозволяють зробити його популярним вибором бази даних для багатьох підприємств [11]. Обрана нереляційна база даних характеризується простотою використання та встановлення, низькими витратами ресурсів та швидкодією. З її допомогою в сукупності з обраним фреймворком можна пришвидшити та спростити влаштування деякого функціоналу, а саме кешування даних, які мають цінність невеликий відрізок часу, використання черг та інший функціонал, який потребує швидкого виконання.

4.1.7 Ubuntu 16.04

Дистрибутив Unix-подібної операційної системи Linux. Є безкоштовним та розповсюдженим програмним забезпеченням. Особливістю Unix-подібних систем є їх термінал. Майже всі програмні пакети, не дивлячись на те, що вони мають графічний інтерфейс, можуть використовуватися через термінал за допомогою спеціальних команд [14]. Термінал можна порівняти з командним рядком операційної системи Windows, хоча схожі вони лише за принципом використання. Командний рядок доволі простий та малофункціональний в порівнянні з терміналом, який в свою чергу може бути застосованим майже для будь-чого [15]. Він надає можливість редагування файлів за допомогою вмонтованого текстового редактора. Також якщо встановити відповідні пакети, через

нього можна запустити браузер, який буде виводити лише вміст сайтів без стилів та html-розмітки. За аналогією, до таких браузерів створюють пакети для використання месенджерів та багатьох інших звичних нам програм.

4.1.8 Програмний пакет ngrok

Даний пакет використовується для тунелювання. Тунелювання являє собою захищений зв'язок між двома кінцевими точками. Існує багато варіанти його використання. Його можуть використовувати безпосередньо для створення захищеного зв'язку між кінцевими точка доступу до локального хосту з мережі Інтернет та інших цілей. В розроблюваному проекті потреба тунелювання пов'язана з умовами використання вебхуків Telegram Bot API, а саме посилання, на яке будуть надсилатися вебхуки, повинне використовувати захищений протокол HTTPS. В свою чергу, даний програмний пакет, при створенні тунелювання, надає посилання на хост як по захищеному протоколу HTTPS, так і по звичайному HTTP.

```
ngrok by @inconshreveable

Session Status      online
Session Expires    7 hours, 59 minutes
Update              update available (version 2.3.29, Ctrl-U to update)
Version             2.2.8
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://3dc84b88.ngrok.io -> localhost:80
Forwarding           https://3dc84b88.ngrok.io -> localhost:80

Connections         ttl    opn    rt1    rt5    p50    p90
                   0      0      0.00   0.00   0.00   0.00
```

Рисунок 4.4 - Інтерфейс програмного пакету ngrok

4.1.9 Пакетний менеджер Composer

Даний пакетний менеджер розроблений для мови програмування PHP. Він використовується для керування залежностями в додатках розроблених на PHP. Застосовується через командний рядок та не має графічного інтерфейсу. Основною задачею пакетного менеджера є підтримка максимальної доступної версії залежностей, таких як бібліотеки, які сумісні з іншими встановленими залежностями. Також з його допомогою можна встановити нову залежність в розроблюваному додатку. Основним джерелом залежностей є офіційні репозиторії, визначені розробниками пакетного менеджера.

4.1.10 Система контролю версій Git

Система контролю версій Git - система контролю версій призначена для спрощення процесу сумісної розробки. Це інструмент для сумісної роботи з файлами. Базується на репозиторіях, які являють собою історію файлової системи в межах папки в якій був ініціалізований репозиторій. Поява Git змінила підхід до розробки та уявлення про системи контролю версій. Це обумовлено тим, що на відміну від схожих систем, дана система зберігає не повний стан файлової структури за будь-який час, а лише первинний її стан та історію змін в файлах. Такий підхід значно полегшив та пришвидшив процес розробки загалом.

4.2 Серверна архітектура

Серверна частина даного проекту - це веб-проект написаний на фреймворкові Laravel. Фреймворк використовує пакетний менеджер Composer. З його допомогою створюється пустий проект, а також, в подальшій розробці та підтримці проекту, використовується для додавання та оновлення залежностей або, як їх ще називають, бібліотек.

В проекті є базово встановлений функціонал для роботи з маршрутизацією. Він надає можливість визначити відносні посилання, а також функції, які будуть відповідати за обробку запитів за цими посиланнями. У загальному випадку, запити на ці посилання можуть надсилати як Front End частина сайту, так і сторонні сервіси, за умови виконання певних вимог. В розроблюваному проекті не реалізована Front End частина, тому в проекті обробляється лише одне посилання, яке використовує Telegram Bot API для надсилання вебхуків. Весь функціонал проекту являє собою обробник вебхуків та допоміжний, для цього обробника, функціонал. З його допомогою генеруються повідомлення для відправки користувачам.

Основною сутністю, яка повертається через вебхук, а також відправляється у відповідь, є повідомлення, тому що інтерфейсом для взаємодії користувача з проектом, є месенджер. Тому розроблено блок визначення типу, отриманого на сервері, вебхуку. Типи вебхуків описані в пункті 3.3.1. Після визначення типу вебхуку, задіюється відповідний блок для обробки даних та генерації відповіді. Після обробки та генерації відповіді, система відправляє згенероване повідомлення користувачеві.

Висновки до розділу 4

У цьому розділі було розглянуто архітектуру серверної частини проекту. Сервер - це віддалена робоча машина, яка не має графічного інтерфейсу. Всі його потужності задіяні для роботи одного або декількох проектів. На сервері встановлені пакети, яких потребує проект, а саме: PHP, Apache2, MySQL тощо.

Серверна частина проекту, це веб-проект, що написаний на фреймворкові Laravel. Весь функціонал проекту являє собою обробник вебхуків та допоміжний функціонал до нього.

Можна зробити висновок, що обрані для ведення розробки технології є сумісними між собою. Загалом, цей набір технологій часто застосовується для розробки веб-проектів, яким є даний проект. Кожна з обраних технологій є сучасною, тобто в процесі розвитку. Отже, весь набір технологій, що застосований у практичній частині проекту, є вдалим.

5 ІНСТРУКЦІЯ КОРИСТУВАЧА

5.1 Процес реєстрації

Першим етапом у користуванні веб-додатком в месенджері є реєстрація. Потрібно знайти Телеграм-бота та зайти в діалог з ним. Знизу екрану діалогу знаходиться кнопка «Запустити». Після її натиснення треба надіслати свій контакт за допомогою відповідної кнопки, яка з'явилась на місці клавіатури (Рисунок 5.1).

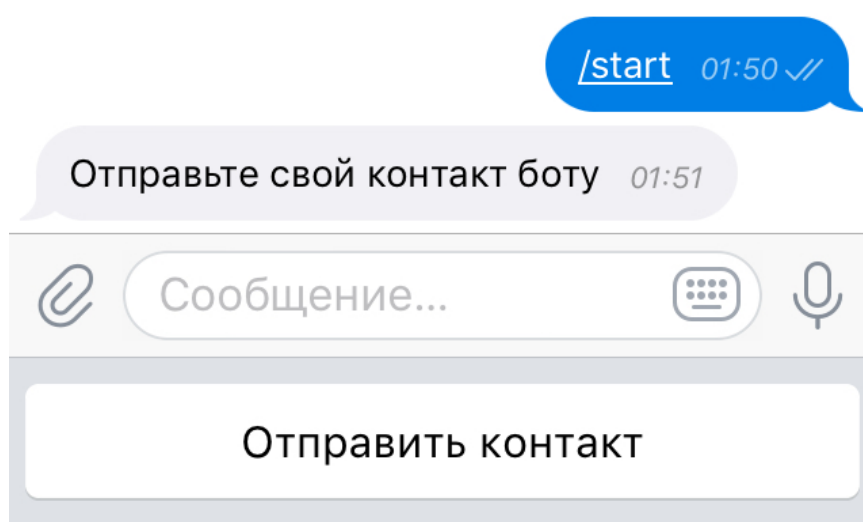


Рисунок 5.1 - Клавiша вiдправки контакту

Після відправки контакту потрібно обрати ким є користувач: студентом чи викладачем. Зробити це можна за допомогою однієї з відповідних клавiш, які прикріплені до повідомлення запитання (Рисунок 5.2).

Далі потрібно обрати свій факультет. Список факультетів буде таким самим, як і вибір ролі в системі, а саме списком, який розміщений під повідомленням із запитанням. В якості тексту клавiш використовуються аббревіатури назв факультетів.

Наступним кроком необхідно обрати спеціальність. Запитання про вибір спеціальності відправляється одразу після вибору факультету. Обрати потрібно із списку представленого під повідомленням з відповідним запитанням.

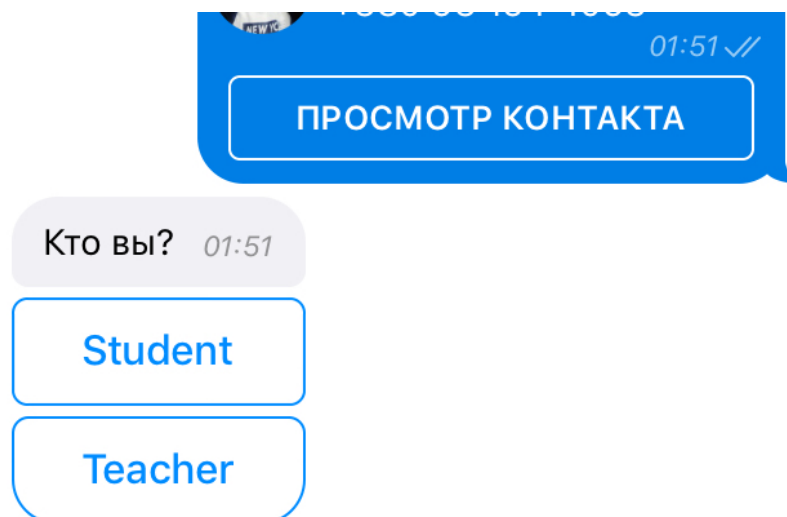


Рисунок 5.2 - Вибір ролі в системі

Після вибору спеціальності, відповідь зберігається, а також відбувається перевірка вибору ролі користувача. Якщо користувач обрав роль студента, наступним кроком буде надіслане питання про вибір групи, в якій він навчається. Якщо було обрано роль викладача, крок вибору групи пропускається.

Далі бот поставить три запитання про реальні ім'я, прізвище та по батькові користувача. Відповідь на них дається самостійно, звичайним повідомленням у полі вводу повідомлення.

Наступним кроком потрібно ввести свій контактний email і написати супровідне повідомлення про причину реєстрації. Вся зібрана інформація буде передана адміністратору для перевірки та підтвердження користувача.

5.2 Додавання матеріалів

Призначення даного функціоналу полягає у наданні викладачами спільного доступу до навчальних матеріалів всім користувачам системи. Додавання матеріалів відбувається у відповідному пункті інтерактивного меню «Додавання матеріалів».

Коли користувач перейде до даного пункту меню, йому буде запропоновано відправити файл, до якого повинен бути наданий спільний доступ. Після відправки файла потрібно надіслати ім'я, яке буде відображатися в списку матеріалів викладача.

5.3 Отримання інформації про користувача

За допомогою даного функціоналу можна отримати публічну інформацію про користувача системи. Для отримання інформації потрібно перейти в інтерактивне меню системи. Після чого перехід до списку факультетів університету. Далі необхідно натиснути відповідну кнопку інтерактивного меню «Факультети». У відповідь буде виведено список кнопок з аббревіатурами факультетів.

Наступним кроком обирається факультет, до якого відноситься шуканий користувач. Після вибору факультету відображається список спеціальностей, які є на даному факультеті у вигляді кнопок з аббревіатурами їх назв.

При обранні спеціальності, до якої відноситься шуканий користувач, потрібно обрати ким він є, студентом чи викладачем. Для цього будуть відображені дві відповідні кнопки. Якщо було обрано студентів, відображається список студентських груп, серед яких потрібно обрати ту, в якій навчається потрібний студент. Обравши групу буде відображений список її студентів.

Якщо було обрано викладачів, буде відображено їх список. Після вибору користувача буде показана інформація про нього, а також можливі способи зв'язку з ним.

5.4 Додавання користувацьких чатів

Додавання користувацьких чатів являє собою функціонал додавання групових чатів для часткового адміністрування ботом. Для цього потрібно створити групу та запросити бота, надавши йому адміністраторські права. Користувач обов'язково повинен бути зареєстрованим та верифікованим в системі.

Після додавання бота до групи потрібно обрати тип чату. Для цього здійснюється перехід в особисті повідомлення, де буде можливість обрати тип створеного діалогу.

5.5 Отримання матеріалів

Після додавання матеріалів викладачами, за допомогою функціоналу описаного в підрозділі 5.2, потрібно отримати ці матеріали. Для того щоб отримати їх необхідно знайти інформацію про викладача за допомогою функціоналу описаного в підрозділі 5.3.

В отриманому розділі меню буде розділ «Матеріали». Перейшовши до цього розділу буде відображений список кнопок, текстом яких є назви матеріалів. Натиснувши на будь-яку з них, у відповідь надсилається відповідний файл.

Висновки до розділу 5

В даному розділі покроково розглянуто функціонал проекту з точки зору користувача системи. Також даний розділ являє собою інструкцію з використання кожного з функціональних елементів, а саме: процес реєстрації з детальним описом, додаванням та отриманням файлів, матеріалів і інформації, створення чатів тощо.

Детальний огляд системи показує, що система є інтуїтивно зрозумілою для користувача. З цього можна зробити висновок, що не дивлячись на об'єм кожної функціональної частини, система залишається зручною для використання.

					ІА52.160БАК.002 ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підп.	Дата		56

ВИСНОВКИ

В наш час, стали популярними месенджери, соціальні мережі, веб-сервіси тощо. Це легкий та зручний спосіб взаємодії, а отже виникла потреба впровадження веб-технологій у навчальний процес.

Завданням дипломного проекту було спрощення взаємодії викладачів та студентів під час навчального процесу. У розділах розглянуто:

- варіанти архітектурних рішень, як окремих частин системи, так і загальну структуру побудови проекту;
- аналоги розроблюваною системи, які в різній мірі підходять до вимог, поставлених перед проектом;
- опис функціоналу, розробленого в системі;
- архітектуру серверної частини проекту з детальним оглядом використаних програмних пакетів;
- використання функціонал проекту в користувацькому інтерфейсі.

Для досягнення мети було вирішено такі задачі:

- проведено аналіз схожих систем, які могли б вирішувати поставлене завдання;
- проведено аналіз технологій для розробки дипломного проекту;
- обрано поєднання технологій для швидкодії та стійкості системи;
- розроблено частину функціонального елемента серверної архітектури сервісу;
- розроблено Телеграм-бота для спрощення взаємодії викладачів та студентів під час навчального процесу.

Детальний огляд проекту показує, що розроблена система взаємодії з користувачем є інтуїтивно зрозумілою для нього. Проект оснащений достатньою кількістю функціоналу, щоб в повній мірі, задовольняти поставлені цілі дипломної роботи. Не дивлячись на об'єм та складність кожної функціональної частини, розроблена система залишається

зрозумілою та зручною у використанні. Проект дає змогу спрощення взаємодії, спілкування, обміну інформацією між викладачами та студентами під час навчального процесу. Завдяки мобільності, доступу у будь-який час, за межами університету, онлайн.

Підбиваючи підсумок, можна сказати, що тему роботи «Телеграм-бот для взаємодії викладачів та студентів» розкрито, а мету дослідження досягнуто.

					ІА52.160БАК.002 ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підп.	Дата		58

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. А. Аюлов. E-learning как новая парадигма в системе высшего образования / А. Аюлов, С. Кенжебулатова / [Электронный ресурс] – Режим доступа до сайту: <https://articlekz.com/article/13079>
2. Telegram Bot API Documentation / [Электронный ресурс] – Режим доступа до сайту: <https://core.telegram.org/bots/api>
3. Compiling and Installing / [Электронный ресурс] – Режим доступа до сайту: <http://httpd.apache.org/docs/current/install.html>
4. Сэмми Пьюривал. Основы разработки веб-приложений / Сэмми Пьюривал. – СПб. : Питер, 2015. – 272 с. – (Ил. Серия «Бестселлеры «O'Reilly»).
5. Томсон Лаура. Разработка Web-сайтов на PHP и MySQL ; [пер. с англ., 2-е изд.] / Лаура Томсон, Люк Веллинг. – СПб. : ООО «ДиаСофтЮп», 2003. – 672 с.
6. Робин Никсон. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Робин Никсон ; [4-е изд.]. - СПб. : Питер, 2016. – 768 с. – (Ил. Серия «Бестселлеры «O'Reilly»).
7. Мэтт Зандстра. PHP: объекты, шаблоны и методики программирования / Мэтт Зандстра ; [пер. с англ., 4-е изд.]. / М. : ООО «И. Д. Вильямс», 2015. – 576 с.
8. Matt Stauffer. Laravel: Up and Ranning. A Framework for Building Modern PHP Apps / Matt Stauffer. – USA : «O'Reilly», 2017. – 795 p.
9. Taylor Otwell. Laravel Documentation / [Электронный ресурс] – Режим доступа до сайту: <https://laravel.com/docs/5.7>
- 10.Святослав Куликов. Работа с MySQL, MS SQL Server и Oracle в примерах / С. Куликов ; [практ. пособ. для программистов и тестировщиков]. – Минск : БОФФ, 2016. – 547 с.
- 11.Редмонд Э. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL ; [пер. с англ. А.А.

Слинкин] / Э. Редмонд, Д. Чилсон. – М. : ДМК Пресс, 2013. – 384 с.
– (под ред. Ж. Картер).

12. Wang Z. Redis 4.x Cookbook / Z. Wang, P. Huang. – Birmingham-Mumbai : OOO Packt, 2018. – 614 p.

13. Redis Quick Start / [Электронный ресурс] – Режим доступа до сайту:
<https://redis.io/topics/quickstart>

14. Арнольд Роббинс. Bash. Карманный справочник системного администратора / Арнольд Роббинс ; [пер. с англ.]. - СПб. : «O'Reilly», 2017. – 152 с.

15. Бреснахэн К. Linux на практике / К. Бреснахэн, Р. Блум. – СПб. : Питер, 2017. – 384 с.

					IA52.160БАК.002 ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підп.	Дата		60

ДОДАТОК А

Лістинг програми

```
public function callback(Request $request)
{
    $data = $request->all();
    Storage::put('test_reg/'
        . Carbon::now()->format('Y-m-d-H-i-s')
        . '.json', json_encode($data, JSON_PRETTY_PRINT)
    );
    $this->saveMessage($data);
    if (($resp = $this->unknownUserCheck($data))['status']){
        $this->disableDialog($resp['from'], $data);
    } else {
        $this->parsingCallback($data);
    }
}

public function unknownUserCheck($data)
{
    $parsedInfo = [];
    foreach ($data as $item) {
        if (is_array($item)) {
            foreach ($item as $key => $value) {
                if ($key == 'from' || $key == 'chat') {
                    $parsedInfo[$key] = $value;
                }
                if (isset($parsedInfo['from']) && isset($parsedInfo['chat'])) {
                    break 2;
                }
            }
        }
    }

    $tUser = TelegramUser::where('user_id', $parsedInfo['from']['id'])->first();
    (new TelegramApiService())->send(json_encode([
        'chat_id' => $parsedInfo['from']['id'],
        'action' => 'typing'
    ]), 'sendChatAction');
    $this->checkUser($parsedInfo['from']);
    $role = 0;
    if (isset($tUser->role)) {
        $role = $tUser->role;
    }
    $type = null;
    if (isset($parsedInfo['chat']['type'])) {
        $type = $parsedInfo['chat']['type'];
    }
    Log::info($type . $role);
    if ($role === 0) {
        return ['status' => true, 'from' => $parsedInfo['from']];
    }
    return ['status' => false];
}
```

Продовження додатку А

```

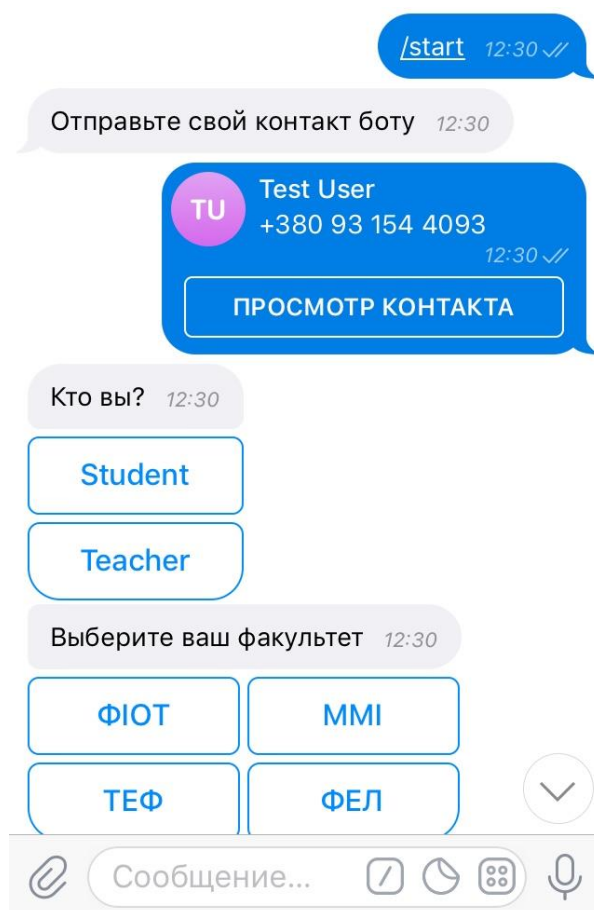
public function parsingCallback($array, $path = '')
{
    foreach ($array as $key => $value) {
        $currentPath = $path;
        switch ($key) {
            case 'message':
                if ($currentPath === '') {
                    $this->saveMessage($array, $key);
                }
                $currentPath = $currentPath . '.' . $key;
                $this->parsingCallback($value, $currentPath);
                break;
            case 'from':
                $currentPath = $currentPath . '.' . $key;
                if (strpos($currentPath, 'callback_query.from') !== false) {
                    $this->menuFnc($array['message']['chat']['id'],
                        $array['data'],
                        $value,
                        $array['message']);
                }
                $this->checkUser($value);
                break;
            case 'entities':
                $currentPath = $currentPath . '.' . $key;
                if (strpos($currentPath, 'callback_query') === false) {
                    $this->checkEntities($array['chat'], $array['text'], $value);
                }
                break;
            case 'callback_query':
                if ($currentPath === '') {
                    $this->saveMessage($array, $key);
                }
                $currentPath = $currentPath . '.' . $key;
                $this->parsingCallback($value, $currentPath);
                break;
        }
    }
}

public function sendNotify($id, $userData)
{
    $t = new TelegramApiService();
    $data = $userData['telegram_contact'];
    unset($userData['telegram_contact']);
    $data['chat_id'] = '607813939';
    $t->send(json_encode($data), 'sendContact');
    $data = [
        'chat_id' => '607813939',
        'reply_markup' => [
            'inline_keyboard' => [[[
                'text' => 'APPROVE USER',
                'callback_data' => 'approve.' . $id
            ]]]
        ]
    ];
    $text = "";
    foreach ($userData as $key => $item) {
        if (in_array($key, ['faculty', 'department', 'group'])) {
            $text .= $key . ': ' . $this->getEntityData($item) . "\n";
        } else {
            $text .= $key . ': ' . $item . "\n";
        }
    }
    $data['text'] = $text;
    $t->send(json_encode($data));
}

```

ДОДАТОК Б

Скріншоти роботи проекту



Зм.	Аркуш	№ докум.	Підп.	Дата

Продовження додатку Б

IA-52	IA-51
IT-83	IT-84

Введіть ваше реальне ім'я 12:30

Test 12:30 ✓✓

Введіть вашу фамилия 12:30

User 12:30 ✓✓

Введіть ваше отчество 12:30

Name 12:31 ✓✓

Введіть ваш контактний email 12:31

test@gmail.com 12:31 ✓

Увага! Введіть правильні дані

Сообщение...

Зм.	Аркуш	№ докум.	Підп.	Дата

IA52.160БАК.002 ПЗ

Аркуш

64


Продовження додатку Б

Test User
+380 93 154 4093
12:32





ПРОСМОТР КОНТАКТА

status: student
faculty: ФІОТ
department: CI
group: IA-52
first_name: Test
last_name: User
father_name: Name
email: test@gmail.com
letter: Because I'm student 12:32

APPROVE USER

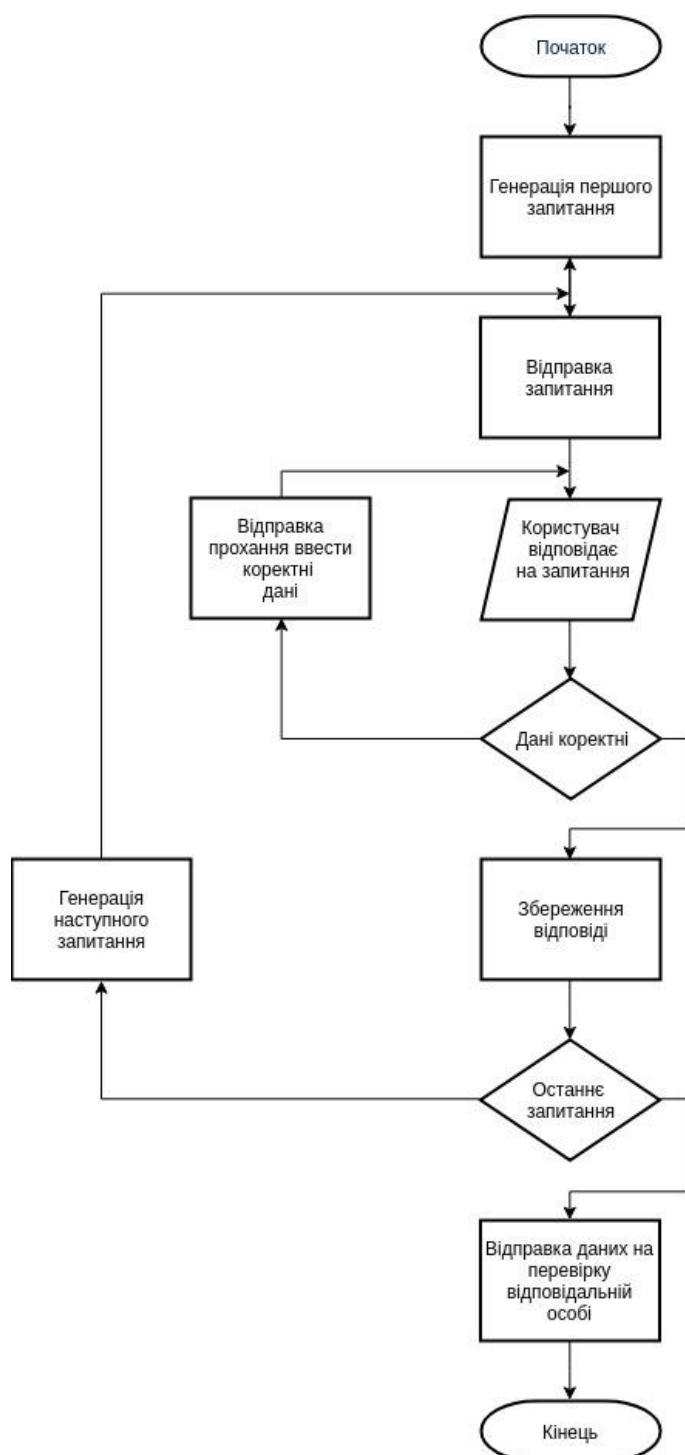


Сообщение...



ДОДАТОК В

Блок-схема алгоритму обробки запитів при реєстрації



ДОДАТОК Г

Блок-схема процесу реєстрації користувача

